REMARKS

Examiner Interview

Applicant thanks the Examiner for the telephonic interview on July 18, 2001, in which the procedural posture of the present application was discussed. Applicant and Examiner Donaghue agreed that the application should be examined as a 37 C.F.R. § 1.53(b)(1) continuation of parent application serial number 08/478,413.

Pending Claims

Due to the uncertainty surrounding the procedural posture of the present application, Applicant is attaching a copy of the claims pending prior to this preliminary amendment as Appendix A.

If the Examiner has any questions or comments, Applicant respectfully requests that the Examiner contact the undersigned by telephone.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Date: July 27, 2001

Sheryl Sue Holloway
Reg. No. 37,850

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025-1026
(408) 720-3476

FIRST CLASS CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage in an envelope addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231

on July 27, 2001
Date of Deposit

Michelle Begay
Name of Person Mailing Correspondence

Signature — July 27, 2001 Date

VERSION OF THE CLAIMS WITH MARKINGS

1. (Amended) A method in a computer system of executing a first sequence of modules in a first task, said first sequence of modules linked to one another and having at least one sequence of execution, comprising[the following steps]:

a. storing in each of said first sequence of modules a skip value indicating a next module in said sequence of modules to execute;

b. executing a first module of said first sequence of said modules; and

c. executing said next module of said sequence of modules indicated by the skip value, wherein each module of said sequence of modules comprises at least one digital signal processing data structure.

4. (Amended) The method of claim 1 further comprising executing [steps] elements a-c on a second sequence of modules, said second sequence of modules being a part of a second task.

10. (Amended) A method performed by a processor of controlling the flow of execution of a first set of executable modules sequentially associated with one another comprising[the following steps]:

a. executing a first module in said first sequence of modules;

b. determining a skip value associated with said first module; and

c. proceeding to execute a subsequent module in said first set of executable modules indicated by said skip value, wherein each module comprises at least one digital signal processing data structure.

18. (Amended) A method of controlling the execution sequence of a series of modules by a processor, each of said modules associated with one another, comprising[the following steps]:

a. executing the first in said series of modules;

b. determining a skip value N stored in said first in said series of said modules;

c. if the skip value N stored in said first module is less than zero, then terminating the execution of said series of modules; and

d. else if the skip value N stored in said first module is greater than or equal to zero then proceeding to a N+1th module in said series of said

modules, wherein each of said modules comprises at least one digital signal processing data structure.

19. (Amended) A method in a computer system of performing a first sequence of modules in a first task, said first sequence of modules linked to one another and having at least one sequence of execution, comprising[the following steps]:

a. storing in a first module of said first sequence of modules a skip value N representing a subsequent module in said first sequence of modules to execute, said skip value N comprising either:

i. an integer less than zero indicating that said first module is a last executable module to be executed in said sequence of modules; and

ii. an integer greater than or equal to zero indicating that said process should proceed to said N+1th module subsequent to said first module in said first sequence of said modules;

b. executing the first of said first sequence of said modules; and

c. executing the subsequent module in said sequence of said modules indicated by said skip value, wherein each module of said sequence of modules comprises at least one digital signal processing data structure.

APPENDIX A

1. A method in a computer system of executing a first sequence of modules in a first task, said first sequence of modules linked to one another and having at least one sequence of execution, comprising the following steps:
   a. storing in each of said first sequence of modules a skip value indicating a next module in said sequence of modules to execute;
   b. executing a first module of said first sequence of said modules; and
   c. executing said next module of said sequence of modules indicated by the skip value, wherein each module of said sequence of modules comprises at least one digital signal processing data structure.

2. The method of claim 1 wherein the skip value comprises the integer N which indicates that execution should skip to the N+1th module following execution of a currently executed module in the first sequence of modules.

3. The method of claim 2 wherein a value of N less than zero associated with the currently executed module indicates that execution of the first sequence of modules should terminate after completion of execution of the currently executed module.

4. The method of claim 1 further comprising executing steps a-c on a second sequence of modules, said second sequence of modules being a part of a second task.

5. The method of claim 4 further comprising the following steps:
   a. storing in each of said second sequence of modules a skip value indicating a next module in said sequence of modules to execute;
   b. executing a first module of said second sequence of said modules; and
   c. executing said next module of said sequence of modules indicated by the skip value.

6. The method of claim 1 further comprising performing a skip action created on the previous iteration of the first sequence of modules.

7. The method of claim 1 wherein the skip value associated with each module in said first sequence of modules may be modified by a module associated with the skip value.

8. The method of claim 1 wherein the skip value associated with each module in said first sequence of modules may be modified by a host associated with the first task.

9. A method of controlling execution flow of a first task comprising a sequence of first executable modules in a processing system by storing in each of said first executable modules a skip count, said skip count comprising an integer N which indicates that execution should skip to the N+1th module following execution of a currently executed module in the first sequence of executable modules, a value of N less than zero associated with the currently executed module indicating that execution of the first sequence of modules should terminate after completion of execution of the currently executed module, wherein each module comprises at least one digital signal processing data structure.

10. A method performed by a processor of controlling the flow of execution of a first set of executable modules sequentially associated with one another comprising the following steps:

a. executing a first module in said first sequence of modules;

b. determining a skip value associated with said first module; and

c. proceeding to execute a subsequent module in said first set of executable modules indicated by said skip value, wherein each module comprises at least one digital signal processing data structure.

11. The method of claim 10 wherein the skip value comprises the integer N, and the subsequent module is the N+1th module following the first module in said first sequence of modules.

12. The method of claim 11 wherein a value of N less than zero associated with said first module indicates that execution of said first sequence of

modules should terminate after completion of execution of said first module.

13. The method of claim 10 wherein the skip value of each module of said first sequence of modules is stored in a datum associated with said each module.

14. The method of claim 10 wherein the said skip value in each of said modules may be modified by a host.

15. The method of claim 10 wherein the said skip value in each of said modules may be modified by each of the respective modules in said first sequence of modules.

16. An apparatus for executing a first sequence of modules in a first task, said first sequence of modules linked to one another and having at least one sequence of execution, comprising:

a. means for storing in each of said first sequence of modules a skip value indicating a next module in said sequence of modules to execute;

b. means for executing a first module of said first sequence of said modules; and

c. means for executing said next module of said sequence of modules indicated by the skip value, wherein each module comprises at least one digital signal processing data structure.

17. An apparatus for controlling the flow of execution of a first set of executable modules sequentially associated with one another comprising:

a. means for executing a first module in said first sequence of modules;

b. means for determining a skip value associated with said first module; and

c. means for proceeding to execute a subsequent module in said first set of executable modules indicated by said skip value, wherein each module comprises at least one digital signal processing data structure.

18. A method of controlling the execution sequence of a series of modules by a processor, each of said modules associated with one another, comprising the following steps:

a. executing the first in said series of modules;

b. determining a skip value N stored in said first in said series of said modules;

c. if the skip value N stored in said first module is less than zero, then terminating the execution of said series of modules;

d. else if the skip value N stored in said first module is greater than or equal to zero then proceeding to a N+1th module in said series of said modules, wherein each of said modules comprises at least one digital signal processing data structure.

19. A method in a computer system of performing a first sequence of modules in a first task, said first sequence of modules linked to one another and having at least one sequence of execution, comprising the following steps:

a. storing in a first module of said first sequence of modules a skip value N representing a subsequent module in said first sequence of modules to execute, said skip value N comprising either:

i. an integer less than zero indicating that said first module is a last executable module to be executed in said sequence of modules;

ii. an integer greater than or equal to zero indicating that said process should proceed to said N+1th module subsequent to said first module in said first sequence of said modules;

b. executing the first of said first sequence of said modules; and

c. executing the subsequent module in said sequence of said modules indicated by said skip value, wherein each module of said sequence of modules comprises at least one digital signal processing data structure.